



# Notice

Voor alle opgaven:

- De limieten zijn beschikbaar op de "Overview" pagina in het wedstrijdssysteem.
- Er is een bijlage die je kan downloaden uit het wedstrijdssysteem die voorbeeldgraders, voorbeeldimplementaties, voorbeeldtestcases en compileer- en uitvoerscripts bevat.
- Je mag maximaal 50 inzendingen maken voor elke taak en je moet in elke inzending precies één bestand insturen.
- Als je jouw programma test met de voorbeeldgrader moet jouw invoer aan het formaat en de randvoorwaarden doen van de opgave, anders kan er onverwacht gedrag optreden.
- In de invoer van de voorbeeldgrader zijn twee opeenvolgende tokens op een lijn gescheiden door een enkele spatie, tenzij een ander formaat expliciet vermeld wordt.
- Als je jouw code op je eigen machine test raden we aan om de scripts in de bijlage te gebruiken. Merk op dat we de `-std=gnu++17` compileroptie gebruiken.
- Als je niet in staat bent om in te zenden naar CMS kan je de `ioisubmit` tool gebruiken om je code op te slaan voor evaluatie na het einde van de wedstrijd.
  - Draai `ioisubmit <task_shortname> <source_file>` in de folder met `<source_file>`
  - Vraag een lid van het comité om een foto van de uitvoer van `ioisubmit` te maken. Jouw inzending zal niet overwogen worden tenzij deze stap gedaan wordt.
    - Als je online meedoet, vraag dan aan je proctor om een foto te maken van de output van `ioisubmit` en stuur die naar de organisatoren.

# Conventies

De opgaven geven functiedefinities met generieke namen `void`, `bool`, `int`, `int[]` (array) en `union(bool, int[])`.

In C++ gebruiken de graders de volgende datatypes of implementaties:

<code>void</code>	<code>bool</code>	<code>int</code>	<code>int[]</code>
<code>void</code>	<code>bool</code>	<code>int</code>	<code>std::vector&lt;int&gt;</code>

<code>union(bool, int[])</code>	lengte van array <code>a</code>
<code>std::variant&lt;bool, std::vector&lt;int&gt;&gt;</code>	<code>a.size()</code>

In C++ is `std::variant` gedefinieerd in de `<variant>` header. Een functie met `std::variant<bool, std::vector<int>>` als type van de teruggegeven waarde kan een `bool` of een `std::vector<int>` teruggeven. De voorbeeldcode hieronder toont drie voorbeelden voor het gebruiken van `std::variant`.

```
std::variant<bool, std::vector<int>> foo(int N) {
    return N % 2 == 0;
}
std::variant<bool, std::vector<int>> goo(int N) {
    return std::vector<int>(N, 0);
}
std::variant<bool, std::vector<int>> hoo(int N) {
    if (N % 2 == 0) {
        return false;
    }
    return std::vector<int>(N, 0);
}
```